Habitat Demonstration Unit Core Avionics Software

Dennis Lawler, *Johnson Space Center*
Lui Wang, *Johnson Space Center*
Tam Ngo, *Johnson Space Center*
Linda Moreland, *Johnson Space Center*
Danny Carrejo, *Johnson Space Center*
Aaron Schram, *Johnson Space Center*
Thomas Matthews, *Johnson Space Center*
Craig Russell, *Johnson Space Center*
Emmy Alex, *Johnson Space Center*
Ayman Qaddumi, *Johnson Space Center*

Increased crew autonomy will be required to support long-duration missions and missions with significant time delay. This will include capabilities such as gradual automation of routine tasks, increasing crew efficiency, the ability to revert to manual mode in the event of off-nominal behavior, and the ability to refine procedures onboard without waiting for ground inputs. A flexible software test bed that allows engineers to exploit such capabilities will be required to productively experiment with these types of advanced operational functionalities.

The multicenter Habitat Demonstration Unit (HDU) project designed and built a planetary surface analog element with a configurable, habitable volume and basic infrastructure services (i.e., power, lighting, command and data handling, thermal control, etc.), as well as a number of "workshop"-like capabilities (e.g., maintenance and repair workstations). This provides an opportunity to implement and operate just such a software test bed. For example, the onboard computer systems furnished a distributed computing environment encouraging new software design approaches, and incorporated wireless data acquisition technology that required new software interface protocols.

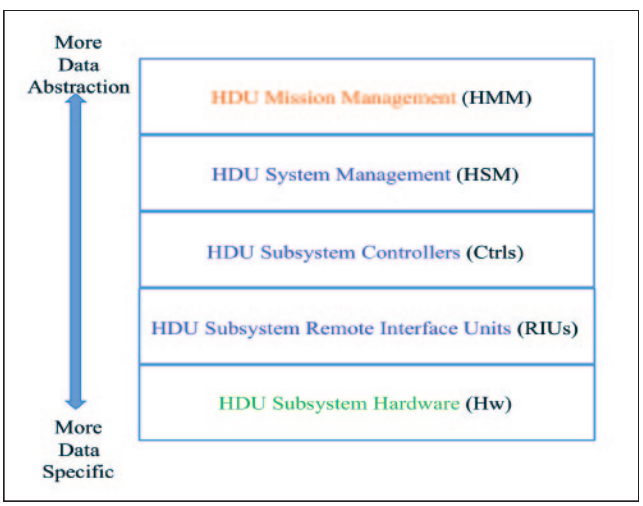

Fig 1. *Functional layers add modularity while providing hierarchical structure.*

The Habitat Demonstration Unit Core Avionics Software (HDU-CAS) is designed to provide the required functionality for an engineering prototype of a highly autonomous space habitat element, and provide an opportunity for new software technologies to be tested in an environment that provides that functionality. As such, the HDU-CAS provides complete command and data handling and intelligent autonomous operations functions to HDU subsystems in all operational circumstances (nominal and off-nominal), and it must do so in a manner that allows technologists to test prototype software for these functions.

Earlier approaches for this problem space relied on manual, one-of-a-kind designs that optimized size and computational performance, often developed unique communication interface protocols, and were deployed on centralized computer assets. Also, they were often highly constrained by the need to integrate with the legacy software environments that restricted the ability to implement new software technology. These approaches would save some time and reduce risk, but were not designed for flexible integration and could preclude any practical experimentation.

In contrast, HDU-CAS was designed and implemented using an innovative mixture of open standard communication products, commercial subsystem diagnostic modeling software, specialized procedural automation scripting software, and a hierarchical structure. It was developed over the period from September 2009 to September 2010, and deployed/evaluated during the Desert Research and Technology Studies exercise in September 2010.

The structure of the HDU-CAS consists of a hierarchical arrangement of hardware interface components (called Remote Interface Units), subsystem-specific controllers, and system management components. The overall component connectivity in the distributed computer network is provided by a message handling middleware between computer assets and a shared memory component with any particular computer (figures 1 and 2).

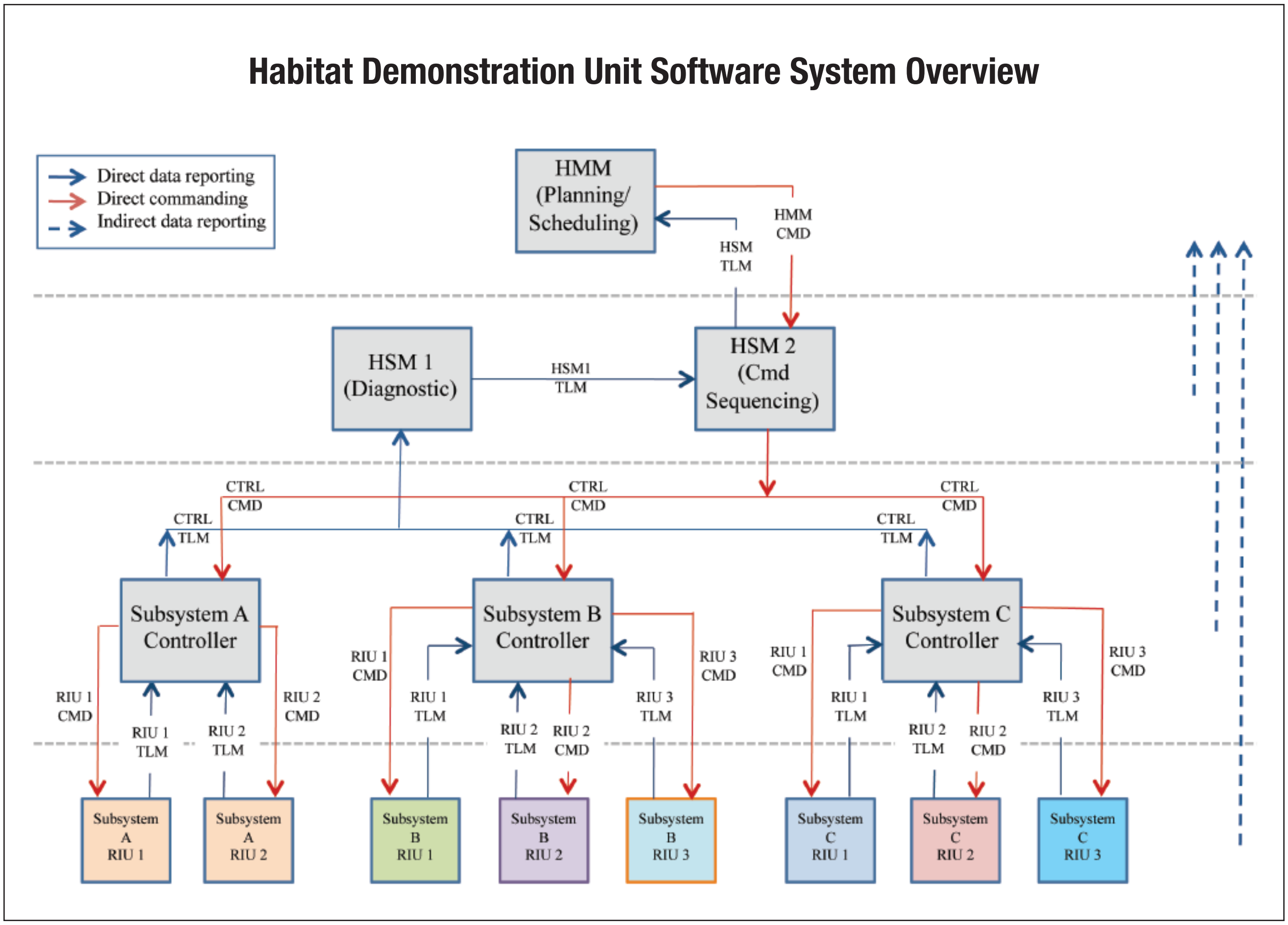

Fig. 2. Middleware connectivity for communication between computers and processes.

The functional operation of the HDU-CAS consists in the use of the data acquisition, logging, message handling, and controller components to monitor telemetry and provide manual and automated command paths for subsystem operations.

The system management components use the collected data and controller observations to manage habitat-wide operations involving the concurrent control of multiple subsystems. Specifically they identify subsystem faults, isolate the diagnosis to problem components, and execute recovery actions either automatically or under the direction of crew/HDU operators via the crew display and interface.

The overall design modeled a structure generally compatible with the ARINC [Avionics Application Standard Software Interface] 653 Safety Critical Software Specification to ease later integration into a deployed space vehicle compliant with that specification.

In addition, previous software environments for spacecraft habitat systems often did not consider *intelligent* automation as a high priority. The HDU-CAS was developed uniquely to integrate technologies that provide intelligent automation and other autonomous operations capabilities right from the start as a high priority.

As deployed for the HDU, this innovative integrated software design provided enormous flexibility and ease of integration that greatly facilitated the dexterity needed when working in such a prototyping environment where subsystem design and behavior can vary broadly as integration proceeds and new operational knowledge is acquired.